

AD-A037 501

HONEYWELL INFORMATION SYSTEMS INC MCLEAN VA FEDERAL --ETC F/6 9/2  
SEMI-ANNUAL PROGRESS REPORT JULY 1975 TO DECEMBER 1975, (U)  
JAN 76 N ADLEMAN, J R GILSON, R J SESTAK F19628-74-C-0193

UNCLASSIFIED

ESD-TR-76-270

NL

| OF |  
AD  
A037501



END

DATE  
FILMED  
4-77

ADA037501

SEMI-ANNUAL PROGRESS REPORT  
July 1975 to December 1975

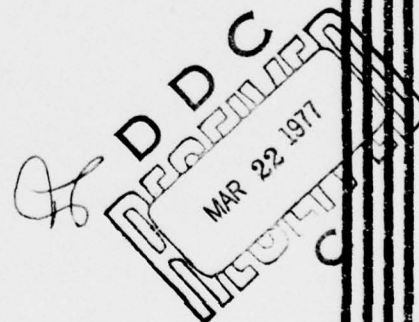
Honeywell Information Systems, Inc.  
Federal Systems Operations  
7900 Westpark Drive  
McLean, Virginia 22101

31 January 1976

Approved for Public Release;  
Distribution Unlimited.

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS  
ELECTRONIC SYSTEMS DIVISION  
HANSCOM AIR FORCE BASE, MA 01731



DDC FILE COPY

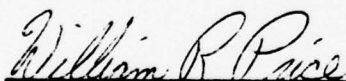
### LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

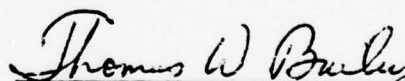
### OTHER NOTICES

Do not return this copy. Retain or destroy.

"This technical report has been reviewed and is approved for publication."

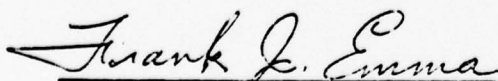


WILLIAM R. PRICE, Captain, USAF  
Techniques Engineering Division



THOMAS W. BAILEY, Lt Colonel, USAF  
Techniques Engineering Division

FOR THE COMMANDER



FRANK J. EMMA, Colonel, USAF  
Director, Information Systems  
Technology Applications Office  
Deputy for Command & Management Systems

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 ESD-TR-76-270	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 SEMI-ANNUAL PROGRESS REPORT July 1975 to December 1975		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) 10 N./Adleman, J. R./Gilson, R. J./Sestak, R. J./Ziller		8. CONTRACT OR GRANT NUMBER(s) 15 FI9628-74-C-0193
9. PERFORMING ORGANIZATION NAME AND ADDRESS Honeywell Information Systems, Inc. Federal Systems Operations 7900 Westpark Drive, McLean, VA 22101		10. PROGRAM ELEMENT, PROJECT, TASK, AREA & WORK UNIT NUMBERS CRDL Item A022
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Command and Management Systems Electronic Systems Division, Hanscom AFB, MA 01731		12. REPORT DATE 11 31 January 1976
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 63p.		13. NUMBER OF PAGES 57
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) DDC MAR 22 1976 ALB		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Security, Security Kernel, Certification, Kernel, Operating System, Multilevel Access, Access Control, Multics		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The goal of Project Guardian is to design, develop, and certify a secure Multics to provide a certified secure multilevel computer utility. The report covers activities from July to December 1975 with an introductory summary of prior work. Activities reported include simplification of the Multics operating system, development of the Multics security kernel, design of the Secure Communications Processor (SCOMP) hardware, development		

409 690

LB  
over



20. of the SCOMP security kernel, methodology for certifying software, and development of ruggedized SCOMP hardware. A list of all documentation produced under the contract during this period is included.

PROJECT GUARDIAN

SEMI-ANNUAL PROGRESS REPORT  
JULY 1975 to DECEMBER 1975

PREPARED FOR  
ELECTRONIC SYSTEMS DIVISION  
L. G. HANSCOM AIR FORCE BASE  
BEDFORD, MASSACHUSETTS 01731  
UNDER CONTRACT NO. F19628-74-C-0193

SUBMITTED BY  
HONEYWELL INFORMATION SYSTEMS, INC.  
FEDERAL SYSTEMS OPERATIONS  
7900 WESTPARK DRIVE  
MCLEAN, VIRGINIA 22101

JANUARY 31, 1976

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	ANAL. and/or SPEC.
A	

## 1.0 INTRODUCTION

### 1.1 Background

### 1.2 Summary of this Progress Report

## 2.0 MULTICS SUPERVISOR REDUCTION

### 2.1 Task Summary

### 2.2 Completed Tasks

### 2.3 Continuing Tasks

### 2.4 Conclusion

## 3.0 SFEP HARDWARE ACTIVITIES

### 3.1 SFEP Hardware Objectives

### 3.2 Technical Approach

### 3.3 Major Accomplishments

### 3.4 Minicomputer Selection

### 3.5 SPM Design

### 3.6 SCOMP Design

#### 3.6.1 Configuration and Interface Considerations

#### 3.6.2 SCOMP Design Performance Analysis Considerations

#### 3.6.3 SCOMP Design Militarization

### 3.7 6000/Series 60 Interface Unit

### 3.8 SCOMP Hardware Verification Methodologies

### 3.9 Future Plans

## 4.0 SECURE MULTICS DEVELOPMENT

### 4.1 Major Accomplishments

## 5.0 SCOMP SOFTWARE DEVELOPMENT

## 6.0 CERTIFICATION ACTIVITIES

### 6.1 The Proposed Environment

### 6.2 The Design

### 6.3 The Suitability of PL/I

## Appendix A Ruggedized Level 6 Computer Development Program

## Appendix B Documentation

## Appendix C References

## PREFACE

This report describes progress under this contract during the period from 1 July 1975 to 31 December 1975.

Under the terms of this contract, Honeywell Information Systems, Inc. is providing the technical integration of the tasks necessary to begin the design, development, and certification of a secure Multics system. Honeywell has subcontracted some of these tasks with Massachusetts Institute of Technology (MIT) and Stanford Research Institute (SRI).

During this reporting period, these tasks included:

1. The reduction in the size and complexity of the software-related functions of the Multics system.
2. The design and development of a Secure Front-End Processor (SFEP) which is based upon a securable minicomputer architecture.
3. The development of specifications for a security kernel for Multics and the SFEP.
4. The analysis of the characteristics of a secure Multics system.
5. An investigation and development of a certification approach for the Multics and SFEP kernels.

MIT performed Task 1 above and SRI performed Task 5 above during this reporting period as subcontractors with Honeywell.

The reader is assumed to be familiar with the Multics system and also the problem of computer security. Related terms such as "Trojan Horse", "Reference Monitor", etc. are not defined in this document.



# SECURITY KERNEL EVALUATION FOR MULTICS AND SECURE MULTICS DESIGN, DEVELOPMENT AND CERTIFICATION

## 1.0 INTRODUCTION

The problem of security in computer systems has been under study for several years. The Air Force has sponsored several studies and development projects aimed at improving understanding of security in computer systems, developing a sound theoretical basis for further work, and demonstrating accomplishments in the field. Many of these projects have been associated with the Multics system.

The overall goal of these efforts has been to develop a certifiably secure computer system for general use by the military to meet their operational requirements. This report describes progress under this contract during the period from 1 July 1975 to 31 December 1975 as part of a long-term plan to take the Multics system from its present form to a prototype secure Multics system which can be used to demonstrate the feasibility of software certification. Three activities are being conducted in parallel to implement this plan. The first parallel effort is the design of a Multics security kernel and the simplification of the Multics operating system. The second parallel effort is the development of a Secure Front-End Processor (SFEP) for integration and certification. The third parallel effort is the development of a technology and a set of tools which allow eventual certification of the Multics and SFEP software kernels.

## 1.1 Background

The military faces an increasing need for operational computer systems capable of processing several levels of classified information at the same time. Present systems are unable to support secure multilevel processing due to fundamental weaknesses in their basic design, since security was not a concern when they were developed. The weakness is that current hardware/software systems are unable to adequately protect the information that they process.

Currently, the military meets the need for processing several levels of information by one of two methods. Either all security levels are processed together at the level of the highest classification present, or each level is processed by itself. Both methods have been less than satisfactory. The problem with processing all levels together is that all users and all equipment, including terminals and communications facilities, must be cleared to the highest classification that the system can ever process. The problem with separate processing is that a separate computer system or a separate period of time is required

for each level handled. Also, sharing of data between users of different clearance levels cannot be permitted. Either method is costly and inefficient. Neither method allows simultaneous handling of information at several levels for users of several levels of clearance.

Multics is the most advanced general utility system as far as security is concerned. Security was one of the initial design goals of the Multics system designers and has been a major concern of the designers and developers throughout the history of the system. Even with this concern for security, the present Multics system cannot be certified secure. Multics, however, does present the best available base upon which to build a certifiably secure multilevel computer utility.

Secure communications has also presented operational problems to the military. A secure on-line system requires a secure communications network. While the techniques of securing communications lines and terminals have been well developed, a certifiably secure communications processor is still undeveloped. A secure multilevel system must have a compatible and secure Front-End Communications Processor to be able to properly handle multiple levels of classified information. Thus the Secure Front-End Processor is essential to the development of a secure Multics.

Both economic and operational considerations make development of a certifiably secure multilevel system desirable. Recent advances in computer technology indicate that it should be possible to produce a system that can process an arbitrary mix of classified and unclassified information simultaneously on a single computer system. The system should serve both cleared and uncleared users and should rely on the computer system's internal hardware/software controls to enforce security and need-to-know requirements. Of primary importance is that the system be certifiably secure. That is, it must be possible to prove that the system is complete and without flaw in any of its security-related aspects.

The Air Force has been working on the problem of providing a certifiably secure multilevel system for several years. In 1970, the Air Force Data Services Center (AFDSC) requested the Electronic Systems Division (ESD) to support development of an open multilevel system for the AFDSC Honeywell 635 systems. The resulting studies pointed out the severity of the problem and led to the formation of a computer security technology planning study panel. The panel's report (1) described the fundamental problems and delineated a program to develop the desired system. The panel recommended that the technical approach to the problem be "to start with a statement of an ideal system, a model, and to refine and move the statement through various levels of design into the mechanism that implement the model system".

The basic component of the ideal system was also identified by this panel. This component is known as the Reference Monitor, an abstract mechanism that controls access of subjects (active system elements) to objects (units of information) within the computer system and enforces the rules of the military security system on such access. Three requirements were recognized for a Reference Monitor:

- a. Complete Mediation - the mechanism must mediate every access of a subject to an object.
- b. Isolation - the mechanism and its data bases must be protected from unauthorized alteration.
- c. Verifiability - the mechanism must be small, simple, and understandable so that it can be completely tested and verified (certified) to perform its functions correctly.

The mechanism that implements the Reference Monitor in a particular computer system has been termed the security kernel. Much subsequent work has been devoted to identifying the characteristics of a security kernel and to exploring the technology involved in producing a security kernel for some computer system.

ESD initiated development of formal mathematical models of the ideal Reference Monitor in 1972. This work (2, 3) resulted in a model of a secure computer system as a finite-state mechanism that makes explicit transitions from one secure state to another. The rules of the model formally define the conditions under which a transition from state to state can occur. The rules have been proven to allow only transitions that preserve the security of information in the system. The model specifies requirements for the operation of a security kernel. These requirements were taken directly from the Defense Department regulations on handling sensitive information (DoD Directive 5200.1-R). With the availability of the model, the problem of validation is now reduced to providing complete assurance that a particular security kernel behaves exactly as the model requires.

Work on the technology of certification progressed in parallel with the work on the model. In 1973, Price (4) identified a methodology for verification of a kernel. More detailed developments of this validation methodology have been reported by MITRE (5, 6). Another approach has been explored which may be more suitable to large software modules (7).

Other activities have been devoted to the problem of building a security kernel for a practical system (8, 9). This work has demonstrated the soundness of the basic concepts and also pointed out some of the problems that lie in the way of realizing a security kernel on a large system. This work has been the basis for development of a secure communications processor which is an



integral component of the long-range goals of this program.

A major project in the development process is the development of a security kernel for a large resource sharing system. The system chosen for this effort is Multics. There are two reasons that this choice was made. First, the hardware base of the Multics system, the Honeywell 68/80 computer, has been identified as best suited of all off-the-shelf large computer systems for the support of a security kernel (10). Second, the Multics system architecture was conceived and developed with security requirements specifically in mind.

One project, now completed, involved the design and production of a Multics system capable of supporting a two-level (Secret and Top Secret) environment for the Air Force Data Services Center (11,12). This system implements security controls based on the military access rules, but it does not completely handle the threat of a hostile penetration. From these efforts, additional insight was gained in the problems of designing and developing a security kernel for Multics.

#### 1.2 Summary of this Progress Report

Design of a security kernel for Multics was started as a joint effort between personnel from ESD, the MITRE Corporation, the Massachusetts Institute of Technology, and Honeywell Information Systems. This design effort has led to a more complete understanding of the general problem and has provided the foundation for a development plan which Honeywell submitted to the Air Force during this reporting period ("Multics Security Integration Requirements, 1 January 1976 - 31 December 1980" as CDRL Item A006 of 31 October 1975).

Work is progressing on formal specifications for the Multics security kernel (13) and on simplification and reorganization of the Multics operating system. Based on these efforts, this report describes the beginning of a major project to refine the current Multics system and reimplement critical portions of the system to produce a certifiable kernel which will interface with a certifiable front-end communications processor with its own security kernel. The result will be a prototype Multics system which may meet the goal of Air Force certification.

The long-range goals of this project can be described in terms of three major development phases: development of a technology to support the development of a certifiable system, development of hardware and software for a prototype Secure Front-End Processor, and development of a certifiable prototype Multics system with a security kernel interfacing with the Secure Front-End Processor.



Due to the complexity involved, these three development phases have been further broken down into five distinct and parallel activities for this performance period as follows:

1. Research into the Reduction of the Present Multics supervisor.
2. Secure Multics Specification Preparation
3. Secure Front End Processor Hardware Development
4. Secure Communications Oriented Processor Software Development
5. Certification Planning

This report describes the progress of these activities during this reporting period. Appendix A describes a hardware development program to ruggedize a Level 6 minicomputer for use as the SFEP. This report then continues with Appendix B by identifying the documentation which has been prepared for the Air Force during this six-month period. Finally, this report concludes with a bibliography.

## 2.0. MULTICS SUPERVISOR REDUCTION

This research phase of the program is being performed by Massachusetts Institute of Technology's Project MAC Computer Systems Research Division as a subcontractor to Honeywell. The specific goals of this continuing research effort are to identify the minimum mechanism that must be correct to guarantee computer enforcement of desired constraints on information access, to simplify the structure of that minimal mechanism to make certification possible, and to demonstrate by test implementation that the security kernel so developed is capable of supporting all the functions of the Multics system. Because Multics permits the direct sharing of information among simultaneous computations, this research can lead to a better understanding of the structures necessary to support the primary Multics functions and, therefore, leads to a Multics system whose security features inspire a high degree of confidence.

At the conclusion of this reporting period, this research project has run for about two and a half years of its intended four year span. So far, the reductions in size and the simplification in structure of the security-sensitive software in Multics that were expected to result from the early tasks is showing significant progress. As the results from these initial tasks are becoming available, and, indeed, being assimilated into the readily available version of Multics, further detailed research is emerging and being contemplated from the experience gained in the early work. Specifically, during this reporting period, a comprehensive study of the Multics Storage system was undertaken by many project participants.

### 2.1 Task Summary

The following is a summary of the status of the various tasks in this research activity.

Prior to this reporting period, the following tasks were completed:

- Removal of the Dynamic Linker from Ring Zero
- Removal of Name Space Management from Ring Zero
- Development of Fast Processes in Ring Zero
- High Level Description of System Functionality
- Study of Removal of User I/O from Ring Zero

During this reporting period, the following research task was completed:

- Formulation of Criteria to Include Modules within the Kernel

Within this activity, the following research tasks are continuing:

- Page Control Restructure
- Traffic Control Restructure
- Answering Service Restructure
- System Initialization Restructure
- Multitasking in the User Ring
- Methodology of Designing a Certified Computer System
- Study of Multics Security Holes
- Restructure of the Network Control Program
- New Buffer Strategy for Input/Output
- Study of Relationship between Reliability and Security
- Study of the Storage Hierarchy
- Support of User-Defined Object Types
- Multics Performance Benchmark
- Independent Domains and Breakproof Services

## 2.2 Completed Tasks

The tasks listed below have been completed by the Computer Systems Research Division of Project MAC at MIT.

### 1. Removal of the Linker from Ring 0

This task was an important first step in pruning unnecessary programs from the portion of the system which must be certified. Several other components of the system, in particular the management of reference names, can be removed from the kernel after the linker has been removed. Final documentation of this task has appeared in the form of a Project MAC Technical Report and a formally presented technical paper (15, 16).

The initial version of the user ring linker showed 7 to 10 percent slower performance than the standard system. This was improved to performance equal to the standard system. The user ring linker will be installed in the standard system as part of a mechanism to prelink the system libraries. This prelinking eliminates a special case in the user ring linker, making it even less complex. This task is now essentially complete as far as Project MAC's Computer Systems Research Division is concerned.

## 2. Removal of Name Space Management from Ring 0

This task removed from the supervisor the facilities for managing the association between reference names and segments in the address space of a process. The association between names and segment numbers is now maintained in the user ring rather than in ring 0, leaving in the supervisor only the association between segment numbers and unique identifier.

Removing the reference name management mechanism from the supervisor required that a data base central to the management of the address space of a process - the Known Segment Table or KST - be split into a private and a common part, and that the supervisor learn to lie convincingly on occasion about the existence of certain file system directories. The result of the removal is a reduction by a factor of five in the size of the protected code needed to manage the address space of a process. Another result is a new simpler interface to the file system portion of the supervisor. Instead of identifying a directory with a character string tree name, a segment number is now used. The algorithm for following a tree name through the directory hierarchy to locate the named element is thus removed from the supervisor. Performance tests indicate that the new design outperforms the old.

The task has been described in a Project MAC Technical Report (17). The mechanism has been combined with the user ring linker and will be installed in the standard system as a unit. This task is now completed as far as the MAC Computer Systems Research Division is concerned.

## 3. Fast Processes in Ring 0

One approach to understanding and simplifying the structure of ring 0 is to separate portions of the supervisor into separate processes. It is necessary that there be available a class of process which is very inexpensive to run so that the separation could be accomplished. This task has involved the design and implementation of such fast processes. A special kind of process has been developed which runs only in ring 0, which has limited capability, and is very efficient to execute.

The fast process is one which makes very restricted demands on its environment. The process has a legitimate stack, can abandon the processor by means of the wait and notify mechanism, and can take page faults, but is restricted from taking segment faults or adding segments to its address space. The wired storage required for these processes has been reduced by two strategies. First, there is one descriptor segment per processor, used by any of these fast processes when it runs. Second, the Process Descriptor Segment (PDS) has been split into two components, only one of which is needed for these processes. Approximately



one-fourth of a page of storage is required for these processes with the rest of the page available for the process stack. Thus if a fast process requires less than three-quarters of a page of stack, there is only one page required in core when that process runs.

Fast processes were tested in use by rewriting the interrupt side of the typewriter device interface module so that it ran as one of these processes rather than directly as a result of an interrupt. Nine pages were able to be unwired as a result of this change. An initial version of this typewriter manager process has run for an extended period on the development machine. The test implementation will not be used in the standard system since a new communication package recently replaced the particular typewriter manager used. Fast processes are currently being installed as part of other tasks, such as the Restructuring of Page Control Task.

This task is essentially completed except for final documentation. Its results will be extensively utilized for further test and evaluation in other tasks.

#### 4. High-Level Description of System Functionality

As part of any attempt to certify a system, it is necessary to have some description of the intended functionality of the system itself to serve as a standard against which to certify. Several members of the project have tried various notational schemes for describing the functionality of various parts of the system. A representation of system data bases and related algorithms in the Vienna Definition Language was performed using the Known Segment Table as a case study. A similar description of directory control, using English as the descriptive language, was also performed. Finally, a language was devised for describing programs with complexity structured data bases, which attempts to avoid implications concerning the implementation of the data base structure. This language is now being used to represent various alternative algorithms being considered as part of the restructuring of page control.

#### 5. Study of the Removal of User I/O from Ring 0

A strategy has been developed for handling user-initiated I/O which operates almost completely in the user ring. The only function which is required within the kernel is the management of multiplexed devices. The scheme uses, as the buffering strategy for I/O, the virtual memory management algorithm of the system. The scheme effectively removes I/O from the kernel of the system, however it requires an I/O controller with capabilities slightly greater than the one currently available on Multics. Thus this particular scheme will not be implemented in the near future. It

is being considered in the further design of the system.

### 2.3 Continuing Tasks

The tasks listed below remain active and will be pursued in the future by the Computer Systems Research Division of Project MAC at MIT.

#### 1. Restructuring of Page Control

Research is continuing on various ways to reorganize page control. Using the language devised under the completed task "High Level Description of System Functionality", a version of page control was constructed which handled read-write sequences in a separate process. This approach was then further refined to produce a version of page control which uses separate asynchronous processes to execute all of the page control functions except the act of fetching the missing page. It is felt that by isolating functions in separate processes, and constraining them by restricting the interprocess communication paths, that it will be easier to understand and certify the overall algorithm. One of the other benefits of structuring page control in this way is that it should be possible for several processors to take and handle a page exception simultaneously, without interfering with each other.

The goal of this task is to utilize several asynchronous parallel processes to perform the functions of page control. Separate processes are used to remove pages from memory and from the paging device so that a free storage pool will always exist to be used for the servicing of page faults. The processes used are examples of the fast processes developed under the completed task "Fast Processes in Ring 0". Use of parallel processes provides simplification of the algorithm, since it eliminates some artificial interactions that occur if the functions are performed as part of the same process and which constrain the functions to run in a particular synchronized order. The new method will also scale up more effectively to a larger system since it eliminates contention on the global page table lock.

Only two steps remain to complete this task. On step, of course, is the final documentation in the form of a technical report. The second remaining step is an investigation of the performance aspects of the new implementation. Initial comparisons between the standard, currently operational Page Control and this experimental version of Page Control suggest that the experimental version requires about one and one half the standard time to process a page fault. It seems that this increase in time results from the experimental version being coded in PL/I and the use of a large number of external subroutine calls which introduced considerable execution time overhead. The true

magnitude of these two differences will be investigated to discover the intrinsic costs of the two algorithms.

## 2. Restructuring of Traffic Control

Techniques are currently being explored to restructure and simplify the traffic controller in order to speed up the act of switching from one process to another and to simplify the mechanisms involved. The intention is to split the traffic controller into two parts, separating out the actual act of switching from one process to another from the more complex act of deciding which process is eligible to run. The division into policy and mechanism should make the algorithm easier to understand.

A design has been proposed to restructure the traffic controller into two levels. The lower level multiplexes the real processors of the system among a fixed number of so called virtual processors. By fixing in advance the number of such virtual processors, this low level processor multiplexor need make no use of the systems virtual memory facilities. Thus there is a strict isolation and ordering between the multiplexor and the virtual memory. A higher level scheduler multiplexes some of the virtual processors among all of the currently operating real Multics processors. This higher level scheduler can use all of the facilities of the Multics virtual memory, since they are implemented at a lower level. It is expected that this restructuring will clarify the relationship between traffic control and page control and also aid in separating the idea of interprocess signaling from the idea of traffic control. In this proposal, no ring 0 data base (such as the current message table) will be needed for messages between processes. Messages between processes will be sent using segments that are protected using the standard system access control mechanisms. This appears to be a great simplification over the current mechanism.

Progress is being made in four areas. First, the low level scheduler which implements virtual processors using real processors is being implemented. Second, the high level scheduler, which will multiplex these virtual processors among real Multics processes is being designed. Third, all portions of the system, other than traffic control, which must be modified or redesigned in order to run the new traffic controller have been identified. Included are modifications to interrupt and fault handling, changes to page fault handling, and various other small system changes. Recoding is in process. Fourth, a proposal has been prepared to eliminate from the system the traffic control data base known as the Interprocess Transmission Table (ITT). This removal would simplify the traffic controller significantly and be another step in the attempt to simplify the various interprocess communication mechanisms being used in Multics.



During this reporting period, no reportable progress was made on this task due to a lack of available manpower.

### 3. Restructuring of the Answering Service

The answering service is made up of those algorithms which authenticate the user, create processes, and manage teletype lines. This is a large interconnected set of functions, all of which are security sensitive given the current modularization. A rearrangement of the answering service has been proposed which will achieve an isolation of those particular components that are in fact crucial to assure secure operation of the system. A similarity has been recognized between the creation of a new process and the entering of a new protection domain. This allows access control lists to be used to regulate the creation of processes on the behalf of any particular user. In general, this scheme avoids the need for certified software by providing the means to assure the user that a process created with the user's identification will start executing only in certain specified programs that the user provides. These programs are provided with tools which allow them to determine that the process has been brought into execution under appropriate circumstances.

During this reporting period, all the code required for the experimental redesign of the Answering Service was completed. The testing and evaluation of the new version is proceeding in parallel with the documentation of the design. Current schedules project that this task will be completed in early 1976.

### 4. Multics System Initialization

If one is to certify that a system works correctly, one must begin by verifying the "initial state" of that system. For this reason it is very important to understand how the Multics system initializes itself. The current initialization system is relatively unstructured and confusing and is apparently not amenable to verification or certification. A proposal has been made for restructuring system initialization that reduces the amount of code in the initialization phase and that simplifies the task of verifying the remainder.

The approach is to recognize that much of what is now considered initialization ought rather to be considered as reconfiguration. Initialization is then decomposed into two phases. The first phase involves getting a minimal Multics up and running. The second phase is a series of reconfigurations based on input describing the actual configuration in use. The advantage of this strategy is that all of the reconfigurations run in a complete, operations Multics environment, which is much easier to understand than a partial and ever changing environment as presented in the various stages of the current initialization



procedure. Also, since the minimal Multics is independent of the actual configuration (given the minimum hardware required), it can be largely generated as the system tape is created. Algorithms which run at the time that the system tape is created are easier to verify since they too run in a fully operational Multics environment.

Coding and design verification of this task significantly progressed during the past six months. Initial verification of the design approach of structuring system initialization as a collection of sequential, dynamic reconfigurations of appropriate hardware and software subsystems is currently being verified. Design documentation is expected in early 1976.

## 5. Multitasking in the User Ring

This task will provide an environment that will allow the user to write various programs as if they were executing in separate cooperating processes. The execution of these processes is supported by multiplexing the one single process of the user. Thus, this is described as multitasking rather than as multiprocessing. The creation of this program environment in the user ring does not directly contribute to simplification of the kernel, but the existence of the facility will allow other modifications that will simplify the kernel. Among these is the design of a simple but effective quit handling mechanism to replace the current complicated mechanism for multiplexing ARPA Network server processes, and in the restructuring of the answering service.

One experiment involved implementation of a version of user ring Interprocess Communication (IPC) as part of an experimental command processor. In this version, the command processor runs each command in a separate task. When a quit is signalled in the user's process, the result is that only one task need be suspended and control returned to the listener. Any new task (command) started will run on a different stack, thus signal handlers for various tasks never become confused and it is possible to restart any task in any order with any number of other tasks suspended. Another experiment involved implementation of an experimental version of the Network server process.

The multitasking environment in the user ring is completed and is usable, but it remains an experimental function requiring further test and evaluation as well as final documentation.

## 6. A Methodology for Designing Computer Systems

This task has been concerned with developing a general methodology for designing certifiably secure systems. The goal is to make the systems easier to certify as correct. A method has been formulated and is being tested by attempting to design a security kernel for a system with functional characteristics similar to Multics. The basic design is complete and some of the abstract programs have been written in the CLU language (which resulted from the completed task "High Level Description of System Functionality"). Work is under way to make the verification of the resulting system easier and on improving the efficiency of the resulting system without making the verification harder.

Documentation of the methodology, design of the Multics-like system, and other results are in process and will appear early in 1976.

## 7. Study of Multics Security Holes

A listing and report is made annually cataloging all known security flaws in the Multics system, ways to violate the security of the system, or ways to crash the system. An attempt is made to analyze each flaw and to identify the general class of problem represented by the flaw.

One general class of security loop-hole was discovered and an automated search was made of the system to determine which modules might be susceptible to this particular attack. Several were found and repaired. After the repairs were made, the nature of the flaw was published to the development community so that the flaw would not be repeated.

This task is a continuing task requiring an on-going review and annual documentation. This documentation has limited distribution by its very sensitive nature.

## 8. Restructuring the Network Control Program

The Network Control Program has represented an ideal candidate to use in an experiment involving a multiple process implementation of a control algorithm, as discussed under "Multitasking in the User Ring". The Network Control Program is concerned with the flow of data to and from the ARPA Network. Its principle function is the management of a multiplexed communication path, which implies the management of multiplexed buffers. It was proposed that the flow of data between these various buffers be implemented using the fast processes in ring 0.

A related area is the possibility that common elements may be identified in the software that is required to handle different multiplexed communication streams. It is possible that there is a similar function in the software that interfaces the ARPANET and the 355. A buffer manager routine is an obvious example of a possible common routine. It is very interesting and profitable to identify these modules and to isolate them.

This markedly reduced the amount of code within the kernel. Also, if all multiplexed communication streams could be handled by one set of kernel modules, such as the network server, this could result in great simplification of the kernel. This observation also has led to the start of a general re-examination of the way that I/O is done by the system.

The restructured Network Control Program was installed during this reporting period. Only final documentation of this task remains.

#### 9. New I/O Buffer Strategy

This task involves the design and implementation of a new I/O buffering strategy which uses the virtual memory itself as a buffer. The task has become part of the task of restructuring the Network Control Program. The task arose from an attempt to increase the efficiency of transmitting data to and from the ARPA Network and at the same time to gain a more basic understanding of the interaction between Input/Output functions and virtual memory computer systems.

The result of this design is a buffer that uses the virtual memory and appears to be infinite in length. The use of the virtual memory eliminates any need to compact or otherwise manage the buffer area, thus reducing overhead. Since the buffer is in the process virtual address space, it is directly accessible to a user process. This avoids the copying of data to make it accessible.

It appears that this I/O buffer strategy can be exploited successfully for all devices for which the system nucleus is responsible. This unification of buffer management is a significant contribution to the certification project due to its reduction of bulk and complexity in the kernel.

This task progressed from the conceptual phase to the design phase during this reporting period in that tape I/O services are being considered through the ARPA Network interface to determine the magnitude of thruput degradation.



#### 10. Formulation of Criteria for Inclusion of Modules Within the Kernel

This task is an attempt to identify a set of general rules which will specify those modules which must be within the kernel of an operating system. One approach is a study of the separation of policy from mechanism in a module.

As part of this task, a discussion of the criteria to be used for including portions of the page control system within the kernel has been produced. The intent is to perform a similar study for various other portions of the system and to attempt to evolve a general theory of the structure of a kernel from them.

This task was completed during this reporting period and final documentation will appear in the "Methodology for Designing Computer Systems" task above.

#### 11. Study of System Error Recovery

The system's ability to recover from errors is related to the problem of system initialization since both must assure or certify that the system is in some known state. This task is interested in determining whether there are some particular structures for data bases and algorithms that make it much easier to assure that the data base is in fact consistent and correct.

This task was combined with the "Study of the Relationships between Security and Reliability" task below during this reporting period.

#### 12. Study of Relationships between Security and Reliability

This task involves studying the relationship between the security of a system and the reliability of that system. In the Multics system, it is presumed that a system failure may have an unknown effect on the security status of the system; this is the reason that the system is shut down, salvaged, and restarted after every unexplained system failure. It is not obvious, however, that security is directly dependent on reliability. If it were possible to determine that certain classes of computer failure could not influence the security state of the system, then the two functions would have nothing to do with each other. More strongly, it is possible that a highly reliable system contains mechanisms that are not desirable from the viewpoint of security. For example, one way to increase the reliable storage of information on a system is to make several copies of that information; many copies, however, increase the probability that the information may be compromised.

In an attempt to understand better the relationship between security and reliability, a study has begun of a variety of systems that maintain high reliability as one of their goals, to investigate mechanisms in the system in the light of their implications for the security of the information stored on that system. The task has been combined with the Study of System Error Recovery task above.

During this reporting period, initial design documentation was prepared and is being reviewed.

### 13. Removal of the Storage Hierarchy from Ring 0

The removal of the linker and of name space management from ring 0 can be considered the first two steps in restructuring the file system. The next logical step is the removal of directories from ring 0. Study has shown, however, that this is not appropriate at this time.

During this reporting period, a comprehensive review of the Multics Storage System was undertaken. This project was led by MIT personnel and was also attended by cognizant Honeywell personnel. This group study reviewed the New Storage System as it was being designed and implemented. Specifically, the Directory Control subsystem and the functions of its related data base, the Active Segment Table (AST), were intensively studied. The purpose of this study was to understand the reasons behind the large bulk and complexity of these facilities and to eventually propose strategies for simplification. It is expected that the output of this study will be a variety of proposals for additional research in various areas related to these topics. One immediate spinoff from this study was the task to separate the Page Control and Segment Control functions within the Active Segment Table to provide further simplification of Page Control. This last task surfaced during this reporting period and progress in the form of initial design considerations was identified late in the period.

It has been proposed that directory control be partitioned into two components, each with its own data base. There will be an unstructured segment catalog, indexed by unique identifier. This catalog maintains the physical attributes for each segment such as file map, bit count, various date-time parameters, etc. There will also be a directory hierarchy which maintains the association between character string names and unique identifiers. This hierarchy will also maintain the access control list for each segment. The principle change in this proposal, relative to earlier proposals, is that the access control list will be stored in the directory hierarchy rather than in a separate access hierarchy.

This task proceeded from task formulation and definition to study, review, and proposal during the past six months. Initial design documentation will be prepared in the near future.

#### 14. Support of User Defined Object Types

A directory can be considered as an object defined in terms of a lower level object, the segment. It is possible that the mechanisms that define the directory could be generalized to allow the definition of new object types defined by users. This sort of ability has been provided in systems that are based on capabilities, but not in systems that are based on access control lists. This task has been considering the question of whether user defined object types can be supported in a system such as Multics.

There are two projects in this area. first, consideration of how extended objects can be supported using an appropriate combination of access control list and capability based mechanisms. Second, the implementation of user defined objects in a pure access control list environment. This project is considering what policies for protection can be imposed upon these user defined extended objects.

Little reportable progress was made on this task during this reporting period due to a lack of available manpower.

#### 15. Multics Performance Benchmark

As the tasks of simplifying the system are accomplished and further modifications to the system are proposed, it is important to be able to determine the performance effects. A stable and reliable performance meter is needed.

Two projects have been underway. One involves rework of the standard Multics performance benchmark. There is now a version of the benchmark that can be debugged on line and is extremely stable in the virtual cpu time required for the run. The other project involves creation of a system load generator. All test load generators which have existed for Multics in the past have suffered from one of two drawbacks; either they use absentee jobs to generate their load (which does not exercise the I/O system) or they require a large number of telephone data sets (which is very expensive). A load generator has been developed that can be run using the ARPANET to drive the test processes. This is useful as it includes a test of the I/O system.

This task progressed during this reporting period in that further refinement and definition of the benchmark contents was implemented.



## 16. Independent Domains and Breakproof Services

This task is exploring some of the implications of removing certain traditional supervisor functions from the Multics security kernel and is exploring an extension of the functionality of the Multics protection mechanisms to allow multiple, independent domains to be part of one process.

A traditional supervisor includes many mechanisms that are not security sensitive simply to protect these mechanisms from accidental damage from user errors. To produce a security kernel for Multics, many such mechanisms are being moved out of the supervisor. By moving them to the user environment, mechanisms such as the linker, the reference name manager, and the search rules become breakable, which could make the system harder to use. Fortunately, the Multics protection rings provide a place to protect non-kernel mechanisms that should be breakproof. They can execute in a ring, say ring 3, above the kernel but below the normal user ring. Because all data bases managed by these service mechanisms are private to a process, they are not part of the security kernel and need not be certified, yet they cannot be broken inadvertently by user errors. Part of this project is the determination as to how to provide such breakproof services for Multics.

The second aspect of this project concerns protected subsystems. Multics has always supported user-defined protected subsystems, although the protection rings can provide only one way protection. It is not possible, however, to use the rings to protect both subsystems and breakproof services at the same time in the same process without making the breakproof services common to all subsystems in a process and therefore part of the security kernel for those subsystems. The essential difficulty is the total ordering of privilege implied by the protection rings. Thus, to provide breakproof services some other way must be found to protect subsystems, if the functionality of protected subsystems is to be maintained. The method being explored is simulating multiple independent domains (containing rings) in a process using multiple descriptor segments for a process. This project is just beginning and progress is continuing.

## 2.4 Conclusion

This section of this progress report has presented the tasks of a large research project to evolve the Multics supervisor into a security kernel which is capable of supporting the functionality of Multics completely and efficiently. The broad objective is finding ways to reduce the size and complexity of the software that must be correct for a shared general-purpose system to be

secure. Reduced size and complexity of security-relevant software is a prerequisite to performing a convincingly logical verification that the system correctly implements the claimed access constraints.

### 3.0 SFEP HARDWARE ACTIVITIES

Current computers utilized for Information Storage and Retrieval (IS&R) and Communications applications are fundamentally incapable of providing adequate security protection for concurrent processing of DoD multilevel-classified information. The current approach of providing either physical separation (multiple facilities), or temporal separation (with intermediate sanitization), is prohibitively costly for the typically large scale systems required.

The Front-End Processors to be utilized with secure large scale IS&R systems and computer utilities require multilevel security in order to extend the security perimeter to include communications.

The solution to this problem is based on security design concepts which are demonstrably sufficient to effect a secure system design which is certifiable and does not prohibitively degrade performance.

This part of the Secure Multics Design, Development and Certification Program encompasses the initiation of the design and development and prototype fabrication plans for a militarized Secure Communications Processor (SCOMP). A specific application has been identified which will demonstrate the functionality of the SCOMP. This application is to function as Secure Front End Processor (SFEP) for a prototype secure machine of the Honeywell Series 60 Level 68 Multics class in a communications network environment. The SFEP will accordingly include the interface unit to the host machine. Also included is initiation of hardware verification plans, development of militarization plans and plans for communications network interfaces.

The design approach for the SFEP is based on concepts derived during the previous phase of this contract. These concepts are in turn based on the Reference Monitor concept developed by the Air Force (and others) (2) which implements the access rules and algorithms of the Bell and LaPadula math-model. (3) The math-model in turn models the access rules of the DoD Information Security System. The math-model is a representation of finite discrete-state mechanisms wherein state-transitions are explicitly governed by rules of the model. Since the math-model is a discrete-state model, it is correlatable to digital computer system architectures.

While the functional requirements of the Reference-Monitor may be performed interpretively (in software), performance and certifiability require that the functions be carefully distributed among hardware and software implementation in order to effect a useful system. The computer architecture selected is



based on the established isolation and mediation mechanisms of the Multics system. The following specific Multics hardware features are noted:

1. Virtual memory system
2. High speed cache for descriptor storage
3. Hierarchical domains (rings)
4. Hardware supported ring crossing

### 3.1 SFEP Hardware Objectives

The objectives of this phase of the 1975 program were to initiate the design, plan for design verification, and plan for fabrication of prototypes of an SFEP for use with the Honeywell 6000/Series 60 computers.

The SFEP effort encompassed five major tasks as follows:

#### 1. Minicomputer Selection

The objective of this task was to select a suitable commercial computer base for the range of applications delineated, which is securable and militarizable.

#### 2. Security Protection Module (SPM)

The objective of this task was to initiate design of an SPM which can be integrated into the selected computer base to add the necessary security controls which will effect a useful secure computer system.

#### 3. SCCMP Design

The objectives of this task were threefold:

##### A. Configuration and Interface Considerations

The objective of this subtask was to plan to augment the SFEP to include communications network interface capabilities.

##### B. Analysis Considerations

The objective of this subtask was to delineate preliminary hardware performance degradations due to the SPM.

### C. Militarization Considerations

The objective of this subtask was to initiate the development of design requirements for TEMPEST and EMC compatibility.

#### 4. 6000/Series 60 Interface Unit Design

The objective of this task was to initiate the design of an interface unit which will permit interfacing the SCOMP to Honeywell 6000/Series 60 large-scale computers.

#### 5. SCOMP Hardware Verification

The objective of this task was to investigate techniques suitable for (a) Hardware Verification; and (b) Deriving a probabilistic measure of security compromise due to hardware failure.

### 3.2 Technical Approach

#### SFEP Functional Design

The SFEP Security approach was to base SFEP Security requirements on the concepts delineated in the Architecture Study Final Report of the previous phase (18). The approach for selection of the minicomputers and the preliminary designs for both the SPM and 6000/Series 60 Interface Unit (IU) were to base them on both the requirements of the Secure Communications Processor Functional Specification developed during the previous phase and the delineated applications (19). The approach to deriving the ultimately recommended specific implementations of the SPM and 6000/Series 60 IU per the SCOMP specification was then based on the selected minicomputer and numerous tradeoffs conducted under Air Force guidance. Detailed functional specifications (Design Specifications, Part I documents) were then developed for both the SPM and 6000/Series 60 IU.

#### SFEP Environmental Design

The SFEP Environmental Design is based on the Honeywell Ruggedized Level 6 computer now in development at Honeywell's Aerospace and Defense Group (ADG). (See Appendix A). This ruggedized Level 6 computer is functionally based on and is compatible with Honeywell Information System's (HIS) commercial line of minicomputers. Five major areas of modifications for ruggedization are listed below:

1. New Chassis Design
2. Circuit Card Stiffening
3. Option of Pin and Socket Connectors at Bus Interface
4. Power Supply Mounting
5. New Control Panel

The designs developed for these modifications are directly applicable to the additional SPM and 6000/Series 60 IO boards and modules required for the SFEP.

The rugged minicomputer will be qualified by Honeywell to its Design Specifications (DS), Part 1. A more detailed description of the rugged minicomputer program is included as Appendix A.

The TEMPEST Design approach developed control plans that provide the design guidance necessary for compliance with RED/BLACK separation and TEMPEST compatibility.

Similarly, the EMC design approach developed control plans that provide the design guidance necessary for compliance with MIL-STD-461A and MIL-STD-462.

### 3.3 Major Accomplishments

The following sections describe the details of the individual Air Force Statement of Work tasks required to initiate the SFEP design. The technical approach to the tasks is described and accomplishments are delineated.

### 3.4 Minicomputer Selection

An analysis was performed which defined the criteria by which a trade study could be done. These criteria were in the form of requirements placed on the commercial minicomputer.

The top level requirements affecting the minicomputer selection are summarized as follows:

#### Functional Requirements

1. The architecture should be bus-structured to support an essentially autonomous Security Protection Module (SPM).
2. The architecture should have basic functionality such that the minicomputer, supported by the SPM, will provide the functionality required for effective



implementation of a Reference Monitor. This includes multiple machine states, fast context switching, suitable address space definition, real-time support and interprocess communication features.

3. The selected computer must have sufficient performance to satisfy the Front End Processor application requirements for the Honeywell 6000/Series 60 computers including Multics.
4. The selected computer must be suitable for a range of communications processor applications.

#### Other Requirements

##### 5. Environmental Requirements

The selected computer should be compatible with, or modifiable to be compatible with, a range of selected physical and electrical military environmental specifications applicable to SFEP and SCOMP applications.

##### 6. Product Support

The selected computer should have continuing corporate product support throughout the useful life of the 6000/Series 60 family of computers.

The selection methodology was to select a candidate set and develop a tradeoff matrix of candidates versus pertinent non-subjective attributes.

It may be noted that many candidates could be summarily eliminated due to criteria 5 and 6 above.

#### Recommendation

The Honeywell Level 6 computers with their ruggedized counterparts were selected as the computers which best satisfy the salient requirements. Within the Level 6 family, the NML-150 was selected as the baseline for the initial SFEP application.

## Task Output

An in-depth discussion of requirements, candidates, tradeoff methodology and selection recommendations is given in the engineering tradeoff study "Design Analysis for a Multics Secure Front End Processor", now in preparation as an ESD Technical Report.

### 3.5 SPM Design

The SCOMP specification is a generic functional specification which defines the functionality required for implementation of a multilevel secure communications processor via Reference-Monitor (2) functionality.

The approach for beginning the design of the SPM was to perform specific (top-level) hardware implementation of an essentially autonomous SPM suitable for integration into the selected minicomputer base and perform implementation tradeoffs. This preliminary design was supported by development of the baseline Detailed Specifications (DS) Part I.

The major tradeoffs considered were as follows:

1. Distributed control logic versus micro-programmed control logic.
2. Mapped I/O versus pre-mapped I/O
3. Long address form versus short address form for virtual address space
4. Several Cache configurations

The tradeoff data was presented at technical interchange meetings and the following implementation decisions were made:

1. Micro-programmed control
2. Mapped I/O
3. Long address form

Based on these decisions and functionality refinements, the DS Part I specification was updated to define the current design requirements of the SPM.

## Task Output

The design specification for the SPM provides a detailed functional definition of the current SPM implementation. This document was submitted to the Air Force as "Security Protection Unit Specification". It is now in preparation for publication as a Configuration Item Development Specification.

### 3.6 SCOMP Design

#### 3.6.1 Configuration and Interface Considerations

The objective of this task was to delineate top level hardware interface and configuration functional requirements for the SCOMP for the following applications:

1. The SCOMP should be capable of simultaneously supporting a variety of terminals in both half-duplex and full-duplex mode at speeds including 110, 134.5, 150, 300, 1200, 2400, 4800, and 9600 bits per second.
2. The SCOMP should support multiple terminals, modularly expandable to a maximum of 256 (although this may require multiple processors working together).
3. The SCOMP must support various external I/O devices.
4. Consideration and planning for the design of brassboard communications network IU's for SATIN IV, Autodin II and the ESD Secure Communications Controller (SCC).

#### Communications Networks

The studies indicate the most cost-effective approach to interfacing to the various communications networks is via the standard product-line communications support modules such as the multiline communications controller (MLCC) and communications line-adapters.

The MLCC is microprocessor-based and thus is programmable and highly adaptable to different line protocols; message text delimiting, editing, and checking; and communications line adapters.

Salient characteristics of the MLCC are as follows:

1. Up to eight full duplex 10.8K bit lines in pairs per MLCC (or two 56K bit lines, or one 72K bit line).



2. User programmable to provide for message delimiting, message editing, and various checking algorithms.
3. Hardware checking (LRC, CRC, etc.).
4. Individual Direct Memory Access (DMA) for each line and transmission direction.

The following line adapters may be utilized for each channel:

1. Asynchronous with RS232-C interface. Speed selectable by software for any of the following bit per second rates: 50, 75, 110, 134.5, 150, 300, 600, 900, 1200, 1800, 2400, 3600, 4800, 7200, 9600.
2. Synchronous with RS232-C interface. Up to 10.8K bits per second including BSC capabilities.
3. Synchronous with MIL-STD-188-C interface. Up to 10.8K bits per second including BSC capabilities (one line per adapter).
4. Direct connect synchronous up to 10.8K bits per second.
5. Broad band synchronous with Bell 301, 303 type interface (one line per adapter).
6. Honeywell Data Link Controller with RS232-C interface (one line per adapter).
7. Bell 801C auto dial.

#### Peripheral Devices

Since the Level 6 computer family is a general-purpose product, a variety of unit-record, random access, and bulk storage peripheral devices are supported. These are suitable for "external" I/O in the 6000/Series 60 context. Peripherals of performance magnitude suitable for Multics "internal" I/O in general are not supported and would require custom controllers.

#### Conclusions

In general the product-supported communications controllers available with the Level 6 computer family will permit direct hardware interfacing to the SATIN IV, Autodin II and ARPA networks with at present only one notable exception. The MLCC provides a 16 bit Circular Redundancy Check (CRC), whereas Autodin requires a 32 bit CRC. This will require some hardware modification.

In addition, depending upon the network entry points, which is application dependent, custom adapter cards and software drivers may be required.

### 3.6.2 SCOMP Design Performance Analysis Considerations

The objective of this task was to initiate performance analyses for the SFEP.

#### Approach

The principle causes of performance degradation due to the SPM were determined to be due to the following:

1. The additional delay in each reference to memory due to the SPM.
2. The time required to load the ultimate SDW in cache upon data descriptor fault.
3. The time required to load descriptor base roots.
4. The additional time required for inter-procedure transfers due to ring crossing barriers.

A comprehensive analysis has not yet been performed since it is both hardware configuration and application dependent. However, several assertions may be made as follows:

#### Memory Access Delay

Assume a delay of 100 or 250 ns delay per memory access (depending upon whether the descriptor was located in Fast Access Store (FAS) or Back Up Storage Cache (BUSC)) and a memory cycle time of 750 ns. Also assume 90% of descriptor hits are in FAS and a typical 60% to 80% of system order time is limited by memory accesses, then a performance loss due to this component would be on the order of 12%. The remaining time delays are application dependent and occur relatively infrequently and are small in comparison.

#### Conclusion

The overall performance, with a degradation limit of no more than 25% as specified by the SCOMP specification, appears readily achievable.

## Test and Evaluation Software

An additional objective of this task was to plan for the eventual test and evaluation of the SFEP hardware utilizing software designed and developed for the purpose of test and evaluation. This test and evaluation software is described in the Test and Evaluation Software Plan, dated January 13, 1976.

### 3.6.3 SCOMP Design Militarization

The major objective under this subtask was to develop initial TEMPEST and EMC design specifications for a militarized SCOMP.

#### TEMPEST

The initial system design effort was the definition of the RED/BLACK requirements. It is considered that a typical situation would find the SCOMP operating in a secure RED area with both high and low speed RED lines and low to medium speed BLACK lines. High speed and low speed are somewhat nebulous terms; in the current context, low speed refers typically to 100-9600 BPS and high speed refers to data rates greater than 50 KBPS. The rationale for the typical situation is as follows:

1. A high speed line is most often used for short distances; that is, beginning and ending within the same controlled area.
2. A typical remote BLACK user would communicate through a channel conforming to MIL-STD-188 or RS-232.

RED users are not precluded from BLACK data; a RED user would obtain BLACK data on a RED line.

Modularity is considered essential if the SCOMP is to fill its multipurpose role. With a bus-structured minicomputer, the I/O capability is achieved via a pluggable I/O card. TEMPEST design problems are minimized if the RED/BLACK modularity is implemented in blocks of one I/O card. Also, maximum flexibility can be achieved by placing the RED/BLACK data isolators in a separate isolator module (or chassis). Thus, only users who require RED/BLACK data isolation would utilize the module. Strictly RED users would not utilize the module.

RED/BLACK isolation and implementation of modularity are further discussed in the Technical Coordination Letter (TCL) No. 2 "SECURE COMMUNICATIONS PROCESSOR (SCOMP) TEMPEST REQUIREMENTS" which was submitted to the Air Force on 19 September 1975. This technical note contained the recommendations that only normal RED/RED isolation be provided between different levels of classified users because it is assumed a properly cleared user



would not act in a covert manner. With the modularity scheme proposed and TEMPEST isolators at the output, any reasonable number of RED users could be isolated to normal RED/BLACK levels.

The system design effort has been documented in the TEMPEST CONTROL PLAN, (Confidential), submitted to the Air Force on 29 October 1975. The TEMPEST CONTROL PLAN is the major output of the TEMPEST test during this program phase. It outlines the TEMPEST program and contains the applicable documents and requirements. It defines the TEMPEST subsystem requirements for the chassis, RED/BLACK isolators, power supply filters, cabling, bonding, grounding, and connectors.

### EMC

A responsible TEMPEST program must be interrelated to the EMC design. The EMC requirements are MIL-STD-461A and MIL-STD-462. The major EMC design constraint is the imposition of TEMPEST, as the two disciplines are interrelated but not necessarily compatible. For example, the power line filter was selected on the basis of meeting TEMPEST requirements, rather than EMC requirements.

An EMC Control Plan was generated outlining a program to achieve EMC compliance in a suitable minicomputer. The EMC Control Plan was submitted to the Air Force as TCL No. 4 on 23 October 1975. As with the TEMPEST plan, it defines system and subsystem requirements, including the chassis, controls and indicators, cabling and connectors. It includes a preliminary Ground and Return diagram which is somewhat peculiar to the preliminary TEMPEST mechanization selected.

### 3.7 6000/Series 60 Interface Unit

The Interface Unit (IU) required for interfacing to the 6000/Series 60 IU is a complicated device with many implementation alternatives in the area of:

1. Entry Port to the 6000/Series 60
2. Number of entry ports
3. Intercommunication rate, channel width, and number of channels
4. Data formats
5. Separation distances accommodated.

Tradeoff data was developed for the various alternatives and presented at Technical Interchange Meetings. (Also see

"6000/Series 60 IU Trade Study", TCL No. 7 which was submitted to the Air Force on 1 December 1975). As a result of the tradeoff data and functionality requirements, the following design decisions were made:

1. Utilize an Input Output Multiplexor (IOM) port in lieu of directly entering the System Control Unit (SCU) of the 6000/Series 60.
2. Utilize the Direct Channel interface to the IOM in lieu of the Peripheral Systems Interface.
3. Design to accommodate, though do not implement a 2000 foot separation between SFEP and 6000/Series 60.
4. Implement single (hardware) communications channel (with software multiplexing).
5. Accomodate greater than 128K words/sec burst transfer rate.

Based on these implementation and functionality decisions, a DS Part I specification was developed which defines the current functionality implementations.

#### Task Output

The Design Specification for the IU provides a detailed functional definition of the current IU implementation. It was submitted to the Air Force as the "6000 IU Functional Specification" and is now undergoing review and revision.

#### 3.8 SCOMP Hardware Verification Methodologies

The major objectives of this Task were to investigate computer hardware verification methodologies applicable to a Secure Communications Processor (SCOMP) and to select recommended techniques which accomplish each verification element. Two major verification elements were identified for analysis. They are:

1. Probabilistic measures analysis of security compromise induced by hardware failure. For this element, the impact of unreliability in the physical hardware on Secure Communications Processor performance must ultimately be analyzed and quantified.
2. Certification that the SCOMP hardware accomplishes the performance requirements of its design specifications. For this element, the hardware certification criteria and methodology for design analysis, design testing, and production product control must be selected and specified.

## Approach

A general investigation of the form and character of available analytic tools and process techniques applicable to hardware verification was conducted. The investigation served to establish the specific tasks appropriate to accomplishing the probabilistic measurement analysis and the certification of the SCOMP hardware design and physical product. Additionally, the range of the available methodologies for each task which should be a candidate for detail study and/or tradeoffs was also determined. A Technical Note on SCOMP Hardware Verification Methodologies which contains descriptions of the work elements necessary to achieve probabilistic measurement and hardware certification and an overview of candidate methodologies was submitted (TCL No. 6, 11 November 1975, "A Technical Note on SCOMP Hardware Verification Methodologies").

The methodology tradeoffs described above were performed and suitable criteria were selected. Where further tradeoffs were inappropriate to a specific task, the task criteria have been developed and specified. These criteria are contained in the appropriate detailed specifications, Quality Assurance Provisions sections, for the SPM and IU.

## Conclusions

The hardware verification methodologies investigation has resulted in recommendations in three areas:

1. Probabilistic measure analysis techniques
2. Hardware design certification techniques
3. Physical product test and certification criteria

A manual probabilistic measures analysis technique was recommended. A SCOMP functional level of analysis was determined to be more suitable than a detail electronic circuit analysis of every component.

A Register Transfer Level (RTL) simulation is recommended to accomplish the hardware design certification. The simulation would encompass the SPM and the portions of the CPU dedicated to support the SPM interface. A similar technique may also be employed for the 6000/Series 60 IU.

Test and inspection criteria were developed and included in the SPM hardware DS Part I specifications. These criteria include reference monitor functional specifications. These criteria include reference monitor functional exercising, electronic parts logical tests for production units, and configuration inspections to insure integrity of the production product.



The design verification report describes in detail all tradeoffs performed and concomitant recommendations. It has been published as "Probabilistic Measures of Compromise", ESD-TR-76-160.

### 3.9 Future Plans

Future plans are to complete the SPEP design; fabricate, test and evaluate prototype SPEPs and support integration into the prototype Secure Multics systems on a schedule consistent with program requirements. A detailed description of future plans, schedules and phasings are delineated in the Honeywell document "Multics Security Integration Requirements" (31 October 1975), Section 11. This document is now in review and preparation for publication as an ESD Technical Report.

#### 4.0 SECURE MULTICS DEVELOPMENT

A secure computer system is one which can successfully protect all data entrusted to it from unauthorized disclosure. This is the basic definition of system security or more specifically system software security which guides the Guardian project. Issues of physical security which can deny service to authorized users are specifically ignored here (e.g., fire, flood, etc.). The major concern is to counter all security threats which would allow someone to steal information (or data) from the computer system. The security threats of general interest fall into three logical areas: malicious persons external to the system, authorized users of the system and collusion between authorized users.

The threats from malicious persons external to the system are not particularly interesting to the system software designer. These threats include: tapping communication lines; stealing listings, tapes, terminal output or other data generated by the system; stealing passwords of authorized users; monitoring electromagnetic emanations from the hardware; or unauthorized actions by operations or administrative personnel. Each of the threats mentioned can only be countered by physical or procedural security measures external to the computer system. The only external threats of interest to the system software designer are illegal attempts to enter the system (login) and operational errors. These are solved by the use of passwords for user authentication and by providing unambiguous instructions and/or messages to operations personnel.

The remaining security threats come from users authorized to enter into and use the system. This is the area of particular interest in this development effort. The less severe internal threats of browsing by a curious user and accidental granting of access have been addressed by the implementation of the Access Isolation Mechanism. The insidious threats of a Trojan Horse program or system penetration remain to be solved.

Within the Multics architecture, a general solution to the threat of a Trojan Horse has not been found. However, for a Trojan Horse program to be able to compromise data, it must be able to communicate between security levels. Therefore, one requirement of this effort is to eliminate all communication paths which would allow a program to read data of one security level and write it where it could be read from a lower security level.

A user who can penetrate the supervisory elements of the operating system may be able to invalidate all the access control mechanisms. A penetration can occur from incorrect implementation of the various protection mechanisms or from a malicious programmer inserting special code sequences to provide a "trap door" into the operating system. Therefore, another requirement of this effort is to verify the correct

implementation of the Multics operating system and to verify that no trap doors exist.

The Multics protection mechanisms are implemented within the most privileged protection ring, ring 0. Unfortunately, there are a large number of programs in ring 0 which are very complex. The interactions between these programs are also complex and often subtle or obscure. In addition, there are no mechanisms to protect programs and data within ring 0 from errors in other programs in this ring. Therefore, any attempt to verify the correctness of the current Multics supervisor as it exists is doomed to failure from the start.

The approach to meeting the requirements is to restructure the current Multics operating system to isolate the primitive mechanisms which implement the security access controls. This will form the reference monitor or security kernel of Multics. The mathematical model of computer security is the criterion used in defining the interface between the kernel and other parts of the system. Good engineering practice requires that the current operating system be molded into the new structure rather than attempting a complete top-down redesign. It is expected that several iterations between top-down specification for correctness proofs and bottom-up design for engineering feasibility will be needed.

#### 4.1 Major Accomplishments

The activities over the last six months have concentrated on the bottom-up definition of the security kernel, external Input/Output (I/O), and the subsequent restructuring of the remaining Multics supervisor functions.

##### Multics Kernel

The Multics security kernel contains all functions which provide access control decisions and all hardware/software mechanisms necessary to support the access control functions. It is these functions which must eventually be certified correct for Multics to be secure. The security kernel is defined to include all Ring 0 software (simplified, of course), all trusted processes, the Central Processing Unit (CPU) hardware itself, the memory addressing hardware, the IOM and channel hardware, internal I/O functions, the SPEP communications interface, and the external peripheral I/O interface. A more detailed description of the kernel functions is being prepared in the Multics Kernel Specification.



## Secure Input/Output Services

The means of providing secure internal I/O functions has caused the greatest concern to the project. The original MITRE proposal of handling all external I/O through the SFEP has been replaced due to unwieldy engineering considerations. The high bandwidth interface requirement needed to support high speed devices and the extra problems of supporting this interface over a distance of 2000 feet was determined to be less practical than our primary alternative. We have chosen to provide high speed peripheral I/O services through the IOM which will have to be slightly modified. This method of supporting I/O is presently being provided within Multics. Some hardware modifications to the IOM have been designed which will show that the IOM and the current software mechanism (ioi) form a complete reference monitor for these I/O functions. The SFEP is still required for handling the external communications I/O functions. It has been determined that the DATANET 6600 (the current communications processor) and the current front-end processor software (Multics Communications System - MCS) cannot be certified. Since, by its nature, a front-end processor must handle multilevel data, the front-end processor kernel must also be certified just like the Multics kernel. The SFEP approach is the only way found to support and provide the environment for certification. The results of this I/O study are being documented in a Technical Coordination Letter (TCL).

## Multics Supervisor Restructure

The simplified security kernel and, to a lesser extent, the changes to handle external I/O have required the restructure of several supervisor functions. The possibility of removing the directory control function from the security kernel is being investigated and appears to be very attractive, as opposed to the approach proposed by MITRE. Message segments have been moved into the security kernel since they contain multilevel data. The impact of this design on the user interface (e.g., mail and interprocess console messages) are being evaluated. Some new administrative mechanisms are being proposed to support I/O device assignment according to the security model. The New Storage System (NSS) design has enabled some new system features to be defined. A most desirable feature is to allow creation of upgraded segments. This may become possible when some proposed changes to the quota mechanism are fully investigated. The changes to the user interface and restructuring of supervisor functions are being documented in the Specification for a Prototype Secure Multics System.

## 5.0 SCOMP SOFTWARE DEVELOPMENT

The objectives of this phase of the 1975 program were to specify a security kernel for the Secure Communications Processor (SCOMP) and to begin the design of this security kernel. The SCOMP security kernel should be general purpose in nature, and a suitable base for additional software to permit application of the SCOMP kernel in environments other than just as the Multics Secure Front-End Processor (SFEP) being developed for this program.

### Technical Approach

The SCOMP security kernel approach was to base the kernel requirements on those requirements outlined in the initial kernel specifications provided by the Air Force. These requirements were then reviewed in light of the Secure Communications Processor Specification prepared by Honeywell and modified to remain in accordance with the SCOMP specifications. Once the requirements have been defined for a general purpose SCOMP kernel, then these requirements will be translated into a SFEP Kernel Top Level Specification.

### Major Accomplishments

The efforts accomplished during this time frame were mainly in training software personnel to understand the security issues as well as understand the work that has been performed by both the Air Force and MITRE in the area of the SCOMP security kernel. The Air Force and MITRE have been developing a security kernel for a general purpose communications application for several years. Honeywell has tried to use this effort as the basis for the SCOMP kernel. The Air Force/MITRE effort has uncovered several key security issues which have not been fully addressed in their effort. Honeywell has been trying to address and resolve these issues. MITRE has also been working on a preliminary definition of the SCOMP kernel. MITRE has provided Honeywell with some technical guidance on the functional aspects of the kernel. Honeywell is now expending its efforts on expanding the kernel functional description into a top level specification.

The Honeywell approach has been to first develop a functional description of the SCOMP kernel. This was required so that the effects of the operating system and communication subsystem could be incorporated at the kernel level. This functional description was then reviewed by the Air Force/MITRE and comments were generated. These comments are now being factored into the functional description document.

In addition, Honeywell personnel are in constant communication with Multics system designers to establish the functional description as well as establish the Multics kernel to SFEP kernel interface.

Once the functional description document has been prepared, then the effort can concentrate on the development of the Top Level Specification. Since the learning curve was longer than anticipated, the effort on developing the functional description document has fallen behind schedule. This has impacted the effort required to prepare a draft of the SCOMP kernel top level specification. For the past several weeks, Honeywell has concentrated its personnel on preparing the specification and will continue to concentrate its personnel on this task until an acceptable SCOMP kernel top level specification is completed.



## 6.0 CERTIFICATION ACTIVITIES

During this reporting period, significant progress was achieved in the development of a methodology to certify the Multics and SFEP kernels. These efforts were performed by Stanford Research Institute (SRI) as a subcontractor to Honeywell. Specifically, SRI prepared three significant documents related to the certification efforts of this program as follows:

1. R. J. Feiertag, Preliminary Modularization of Multics (to appear as a Technical Coordination Letter on 8 January 1976).
2. R. J. Feiertag, PL/I as a System Programming Language for a Certifiable Multics (to appear as a Technical Coordination Letter on 27 January 1976).
3. K. N. Levitt and P. C. Neumann, An Interactive Environment for the Specification, Implementation, and Certification of Multics Security Kernels.

### 6.1 The Proposed Environment

On the basis of the work to date, the Interactive Environment will be highly supportive of the effort to certify the Multics security kernel, assuming that 1) the modularization of the revised Multics design is suitable and that 2) suitable modifications are made to PL/I to make it appropriate for certification of the implementation of the environment. The two Technical Coordination Letters listed above both give significant promise that these assumptions can be realistically assured. Also, the development of the environment is realistic on the desired time scale, in that it is based on existing prototype tools and on a carefully developed specification language (14), both of which are currently being used in SRI's secure operating system development for the Department of Defense with considerable success.

The proposed environment includes a data base for keeping track of different modules of the system as they evolve, including their status with regard to specification, implementation, and proof. It also includes facilities for checking syntactic and (to some significant extent) semantic consistency of the specifications. These tools are all based on existing working prototypes. In its ultimate form, the environment can also include tools for formal testing and semi-automatic proofs of correctness. Note, however, that although these latter tools are only partly anticipated by existing prototypes, they need not be considered essential to the verification effort. Nevertheless, even their partial development could be extremely helpful.

The environment will permit effective incremental proofs of the evolving versions of Multics once the first certified version of Multics has been attained, providing an indication of just which steps may have to be reexamined by new proofs. It also will provide significant help in the development process, from the very beginning. The use of the methodology (and the specification language supporting it) should considerably enhance security throughout the design, implementation and proof stages. Furthermore, the approach is immediately extendable to the proof of security (and other properties) concerning subsystems and other parts of the system outside of the kernel.

## 6.2 The Design

Certification of security will be feasible only if the design structure is appropriate. A conclusion of the Preliminary Modularization of Multics note above is that although the existing design is fairly well modularized, it is not hierarchical. The fact that much functionality in the present system does not belong in the Multics Ring Zero is being addressed by the MIT research.

## 6.3 The Suitability of PL/I

The existing PL/I language is deficient in several respects, with regard to certifiability of the Multics security kernel. Reference 2 (above) proposes specific changes to the language, some involving the elimination of features, others involving the restriction of existing language constructs.

Significant recommendations involve strong typing of pointers, avoidance of "unspec", of mismatched declarations, and other nonstandard conversions, restrictions on labels and nonlocal GOTO's, protection of data and procedures between different hierarchical levels, elimination of pictures and PL/I I/O, and associated built-in functions.

APPENDIX A

RUGGEDIZED LEVEL 6  
COMPUTER DEVELOPMENT PROGRAM

(RNML)



## I. INTRODUCTION

The objective of the Honeywell funded RNML development program is to ruggedize the commercial (HIS) Level 6 minicomputers for military and non-benign commercial environmental applications. The ruggedized computer is the hardware base for development of the SCOMF.

In order to contain costs within specified guidelines, the computer design is oriented primarily toward ground, ground-mobile and transport aircraft applications. Primary design effort is directed as follows:

1. New chassis structure
2. Circuit Card stiffening
3. Pin and Socket circuit card connector changes for the bus interface
4. Power supply mounting
5. New control panel structure

## II. DESIGN APPROACH

### Chassis Design

An important consideration of a militarized or ruggedized piece of equipment is the ability of the mechanical structure to withstand vibration and shock levels typical of a military environment. Therefore, the Level 6 computer ruggedization program involves the use of a precision investment casting as the housing for the circuit boards. The relatively large size of the chassis container coupled with intricate external rib patterns and good dimensional accuracy capability of the precision investment casting process, are combined to produce a design that is both cost-effective and sufficiently rugged to meet a variety of military service environments.

Figure 1 depicts the RNML 10 board chassis design.

BEST AVAILABLE COPY

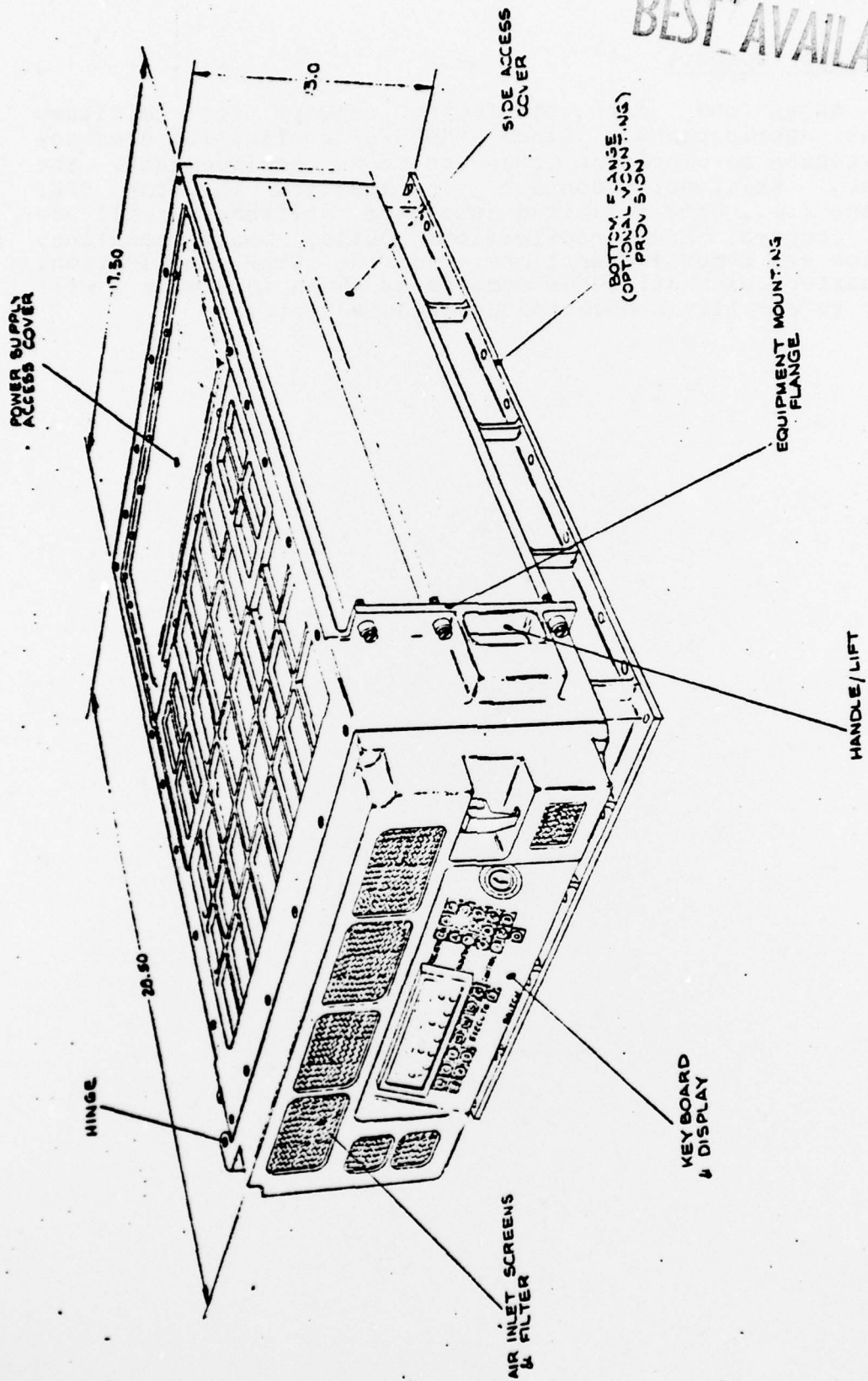


FIGURE 1.  
RNML CHASSIS ASSEMBLY

### Board Stiffener Concept

Figure 2 shows the board stiffening concept for military environment applications. Since the SFEP application does not involve exposure to vibration or severe shock environments, the RNML board stiffener concept is modified for the SFEP application; i.e., only a limited number of stiffeners will be used to control board deflections during board handling, installation and removal operations. For the SFEP application, the two quarter-point stiffener members as shown in Figure 2 will be deleted to simplify production and reduce costs.



BEST AVAILABLE COPY

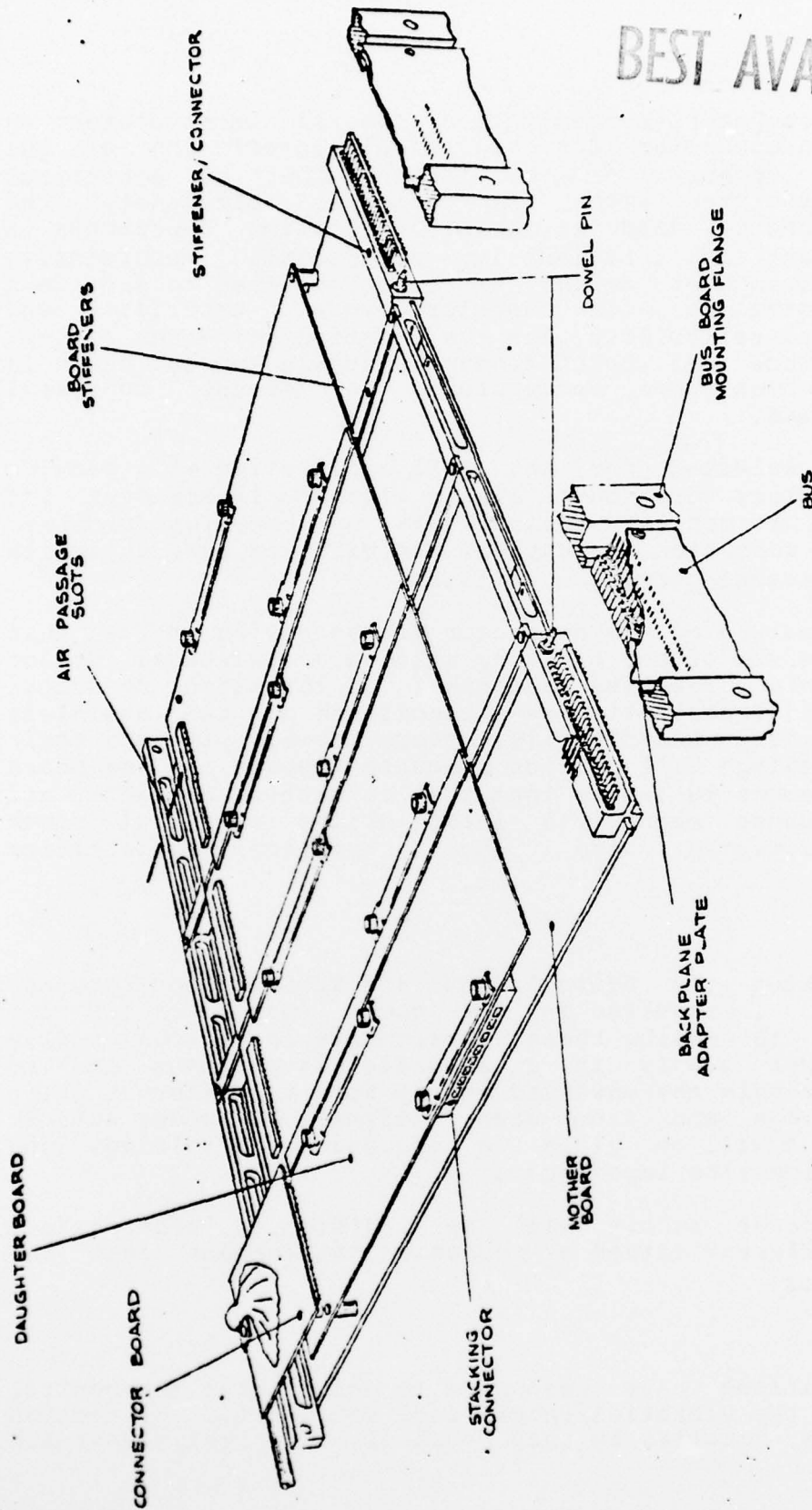


FIGURE 2.  
BOARD STIFFENING CONCEPT

### Circuit Board Connectors

The Level 6 computer ruggedization program incorporates an improved plug-in connector for electrical interfacing of the motherboards. Because of the large number of electrical connections associated with the computer backpanel, the electrical connector used in this application represents a critical component from a reliability standpoint. Accordingly, the basic minicomputer design has been modified to provide a connector mechanization with superior dynamic capability and greater long-term reliability than the existing card-edge system. The proven "blade and fork" connector design concept shown in Figure 1 has been used successfully on several Honeywell Aerospace programs.

The connector selected for the RNML application will provide improved reliability for both static (ground/laboratory) and dynamic (aircraft/mobile) environments by reducing problems associated with corrosion, oxidation, humidity and dust typically encountered in actual service conditions.

A significant feature of the connector mechanization is that this approach permits use of the existing multilayer backpanel design (without change). Positive alignment of the mating connector pins during board installation is accomplished by two stainless steel dowel pins. Additionally, these dowel pins and their mating nylon bushings will provide adequate support for the board and stiffener member to insure that the connector contacts are not stressed under mechanical loads during potential shock conditions associated with bench handling/transportation environments.

### Power Supply

In order to meet the overall EMC and TEMPEST requirements, special attention is required in the power supply area. The basic approach to meeting these requirements is to mechanically isolate the power supply in a separate compartment and to electrically decouple the unwanted energy from the external prime power input lines and from each internal DC power source. Special attention will be given to grounding, shielding, and power transmission line impedances.

The existing power supply will be modified to incorporate a structurally different method of mechanical attachment into the main RNML chassis.

### Control Panel

Design modifications have been made to ensure that the control panel satisfies the vibration/shock and EMC/TEMPEST protection requirements as detailed in this proposal. The design approach

for the control panel involves the use of a separate precision investment casting as the primary structural element for the panel. Suitable RFI sealing gaskets and screened-shielded air inlet openings will be incorporated to provide specification performance for the EMC/TEMPEST requirements.

#### Environmental Design Capability

The details and specifications of Table 1 establish the intended design capability for the RNML equipment. A more detailed description of the RNML is provided in Honeywell Document No. DS-BG8249A1 "Ruggedized Level 6 computer (RNML)", December 15, 1975.



TABLE 1  
ENVIRONMENTAL DESIGN

<u>ENVIRONMENT</u>	<u>SPECIFICATION</u>	<u>REMARKS</u>
Operating Temperature	MIL-E-4158 MIL-E-16400	0 deg. C to +52 deg. C 0 deg. C to +50 deg. C
Non Operating Temp.	MIL-E-4158 MIL-E-16400 MIL-E-5400	-62 deg. C to +52 deg. C -62 deg. C to +75 deg. C -62 deg. C to +85 deg. C
Vibration	MIL-E-4158 MIL-E-16400 MIL-E-5400 MIL-E-5400	2g Peak, Hard-Mounted  (Curve 2g Peak, Hard-Mounted IIA) (Curve 10g Peak, with IA) Isolators
Shock	MIL-E-4158 MIL-E-5400 MIL-E-16400	15g's, 11 ms With Isolators
Humidity	MIL-E-4158 MIL-E-16400 MIL-E-5400	
Altitude (Operating)	MIL-E-4158	0-8000 Feet
Salt Spray Test	MIL-E-16400	
Electromagnetic Compatibility	MIL-STD-461	
TEMPEST	NACSEM 5100	As modified by DCA Circular 370-D195-2

### III. RNML QUALIFICATION TESTS

The RNML will be subjected to inspection and proof tests as defined in Honeywell's Qualification Test Plan, dated January 31, 1976, to verify that the RNML and its components meet the intended specification requirements. The RNML Qualification Unit will be fabricated and inspected to Engineering Released Drawings.

A more detailed description of the Qualification Tests to be performed is provided in Honeywell's "Qualification Test Plan", QTP-BG8249A1, January 31, 1976.

APPENDIX B

DOCUMENTATION

This appendix lists the major documentation that were produced during the past six months. Due to the imminent availability of several significant reports and specifications in January 1976, additional documentation currently in preparation is also provided.

Many other internal notes and working documents were also developed during the last six months but are not included in the following lists.



### Technical Coordination Letters (TCL)

During this reporting period, Honeywell initiated the Technical Coordination Letter (TCL) series of documents. These technical notes communicate important technical issues and efforts as they appear during technical review, investigation, and working meetings. The TCLs that are expected to be available in January 1976 are also listed.

<u>TCL No.</u>	<u>Date</u>	<u>Title</u>
TCL-1	19 Sep 1975	Use of the Honeywell TCL
TCL-2	19 Sep 1975	SCOMP TEMPEST Requirements
TCL-3	19 Sep 1975	SPM Specification - Preliminary
TCL-4	23 Oct 1975	EMC Control Plan
TCL-5	27 Oct 1975	Meeting Minutes - SFEP Technical Working Meeting - 24 Sep 1975
TCL-6	11 Nov 1975	SCOMP Hardware Verification Methodologies
TCL-7	1 Dec 1975	6000/Series 60 Interface Unit Trade Study
TCL-8	12 Dec 1975	Meeting minutes - SFEP Technical Interchange Meeting - 5 Nov 1975
TCL-9	19 Dec 1975	Draft of SPM Design Specification, Part I
TCL-10	19 Dec 1975	Draft of 6000/Series 60 Interface Unit Design Specification, Part I
TCL-11	6 Jan 1976	Meeting minutes - SFEP Software Technical Interchange Meeting - 9 December 1975
TCL-12	19 Jan 1976	Meeting Minutes - SRI Activities - Technical Interchange Meeting, 15 January 1976
TCL-13	27 Jan 1976	PL/I as a System Programming Language for a Certifiable Multics
TCL-14	27 Jan 1976	Preliminary Modularization of Multics
TCL-15	31 Jan 1976	Initial Description of Multics I/O

### Contract Data Items

During this reporting period, numerous technical reports and specifications were submitted to the Air Force for review and approval. The following is a list of these Data Items (CDRL's) and their respective status. Also included are those CDRL's that will be submitted to the Air Force in January 1976.

<u>CDRL No.</u>	<u>Date</u>	<u>Title</u>
A001	15 Oct 1975	Quarterly Report for the Period, July 1975 - September 1975
	15 Jan 1976	Quarterly Report for the Period, Sep 1975 - Dec 1975
A002	15 Aug 1975	Monthly Report for July 1975
	14 Sep 1975	Monthly Report for August 1975
	10 Nov 1975	Monthly Report for October 1975
	10 Dec 1975	Monthly Report for Nov 1975
A004	12 Sep 1975	Interim Report (Draft)
	27 Jan 1976	Interim Report (Revised Draft)
A005	12 Sep 1975	Final Report (Draft)
	31 Jan 1976	Final Report (Revised)
A006	31 Oct 1975	Multics Security Integration Requirements - 1 January 1976 to 31 December 1980 (Draft)
A007	31 Oct 1975	Effects of a Multics Security Kernel (Draft)
A008	31 Jan 1976	Multics Kernel Specification (Draft)
A013	31 Jan 1976	Secure Multics Specification (Draft)
A014	30 Sep 1975	Security and Integrity Procedures (Draft)
	14 Dec 1975	Security and Integrity Procedures (Revised Draft)

## APPENDIX C

### REFERENCES

1. James P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51, October 1972.
2. R. Schell, P. Downey, G. Popek, Preliminary Notes on the Design of a Secure Military Computer System, MCI-73-1, January, 1972.
3. D. E. Bell, L. J. LaPadula, Secure Computer Systems, ESD-TR-73-278, The MITRE Corporation, Bedford, Mass.
4. W. R. Price, Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems, PhD Thesis, Carnegie-Mellon University, June, 1973.
5. E. L. Burke, Synthesis of a Software Security System, MTP-154, The MITRE Corporation, Bedford, Mass.
6. D. E. Bell, E. L. Burke, A Software Validation Technique for Certification: The Method, ESD-TR-75-54, The MITRE Corporation, Bedford, Mass, December, 1973.
7. L. Robinson, P. G. Neumann, K. N. Levitt, A. Saxena, "On Attaining Reliable Software for a Secure Operating System", 1975 International Conference on Reliable Software, Los Angeles, Ca, April, 1975.
8. W. L. Schiller, Design of a Security Kernel for the PDP-11/45, The MITRE Corporation, Bedford, Mass, December, 1973.
9. W. L. Schiller, The Design and Specification of a Security Kernel for the PDP-11/45, The MITRE Corporation, Bedford, Mass, March, 1975.
10. L. Smith, Architectures for Secure Computing Systems, ESD-TR-75-51, The MITRE Corporation, Bedford, Mass, June, 1974.
11. J. C. Whitmore, A. Bensoussan, P. A. Green, A. M. Kobziar, J. A. Stern, Design for Multics Security Enhancements, ESD-TR-74-176, Honeywell Information Systems, Inc., 1974.
12. Access Isolation Mechanism Pre-Release Documentation, Honeywell Information Systems, Inc., McLean, Virginia, August, 1975.
13. W. L. Schiller, P. T. Withington, J. P. L. Woodward, The Top Level Specification of a Multics Security Kernel, Working

Paper WP-20810, The MITRE Corporation, Bedford, Mass., July, 1976.

14. O. Roubine, B. Mont-Reynaud, and L. Robinson, SPECIAL (Specification and Assertion Language) Reference Manual, SRI Memo (to be published in January 1976).
15. P. A. Janson, "Removing the Dynamic Linker from the Security Kernel of a Computer Utility", S.M. Thesis, Department of Electrical Engineering and Computer Science, MIT, Project MAC Technical Report MAC-TR-132.
16. P. A. Janson, "Dynamic Linking and Environment Initialization in a Multi-Domain Computation", ACM 5th Symposium on Operating Systems Principles, Austin, Texas, November, 1975.
17. R. G. Bratt, "Minimizing the Naming Facilities Requiring Protection in a Computer Utility", S.M. Thesis, Department of Electrical Engineering and Computer Science, MIT, July, 1975, Project MAC Technical Report MAC-TR-156.
18. J. R. Gilson, J. E. Mekota, "Secure Communications Processor Architecture Study", (in preparation as an ESD Technical Report), Honeywell Information Systems, Inc.
19. R. Broadbridge, "Secure Communications Processor Specification", (in preparation as an ESD Technical Report), Honeywell Information Systems, Inc.